

2

DTIC FILE COPY

AD-A211 797

REPORT ON ACTION ITEMS FOR
RANGE DEPENDENT MODEL IMPLEMENTATION

DTIC
ELECTE
AUG 30 1989
D & D

SAIC-89/1431

October 5, 1988

SAIC
Science Applications International Corporation
An Employee-Owned Company

Prepared by
Laurie Gainey
Eleanor Holmes
DeWayne White

DISTRIBUTION STATEMENT A

Approved for public release
Distribution Unlimited

Prepared for
Ed Chaika
ONR Code 125

Contract No. N00014-86-D-0137
Delivery Order 21

1710 Goodridge Drive, P.O. Box 1303, McLean, Virginia 22102 (703) 821-4300

Other SAIC Offices: Albuquerque, Boston, Colorado Springs, Dayton, Huntsville, Las Vegas, Los Angeles, Oak Ridge, Orlando, Palo Alto, San Diego, Seattle, and Tucson

89 8 29 052

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. Agency Use Only (Leave blank).		2. Report Date. October 5, 1988		3. Report Type and Dates Covered. Contractor Report October 5, 1988
4. Title and Subtitle. Report On Action Items For Range Dependent Model Implementation			5. Funding Numbers. Program Element No. 63785N Project No. R00120 Task No. Accession No.	
6. Author(s). Laurie GAiney, Eleanor Holmes, DeWayne White				
7. Performing Organization Name(s) and Address(es). Science Applications International Corporation 1710 Goodridge Drive McLean, Virginia 22102			8. Performing Organization Report Number. CR N00014-86-D-0137	
9. Sponsoring/Monitoring Agency Name(s) and Address(es). ONR DET Code 125 SSC, MS 39529-5004			10. Sponsoring/Monitoring Agency Report Number.	
11. Supplementary Notes.				
12a. Distribution/Availability Statement. Approved for public release; Distribution is unlimited.			12b. Distribution Code.	
13. Abstract (Maximum 200 words).				
14. Subject Terms. PARABOLIC EQUATION, MODELING, OAML			15. Number of Pages.	
			16. Price Code.	
17. Security Classification of Report Unclassified	18. Security Classification of This Page Unclassified	19. Security Classification of Abstract.		20. Limitation of Abstract.

REPORT ON ACTION ITEMS FOR
RANGE DEPENDENT MODEL IMPLEMENTATION

SAIC-89/1431

October 5, 1988

Prepared by
Laurie Gainey
Eleanor Holmes
DeWayne White

Prepared for
Ed Chaika
ONR Code 125

Accession For	
NTIS CRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	

Contract No. N00014-86-D-0137
Delivery Order 21



SCIENCE APPLICATIONS INTERNATIONAL CORPORATION

1710 Goodridge Drive
P.O. Box 1303
McLean, VA 22102
(703) 821-4300

SAIC
An Employee-Owned Company

Table Of Contents

	Page
Introduction	1
Section 1: PE Item 1e	2
Section 2: PE Item 1f	9
Section 3: PE Item 1g	13
Section 4: ASTRAL Item 1c	16
Section 5: ASTRAL Item 1d	17
Section 6: ASTRAL Item 1f	19
Appendix A: ASTRAL Programming Notes	22

Introduction

In a meeting at COMSUBDEVRON TWELVE on September 8, 1988, several action items were identified as necessary to the implementation of the Navy Standard Range Dependent models in the Submarine Fleet Mission Program Library (SF MPL). The AEAS program was given responsibility for five of these action items, specifically items 1c through 1g.

All AEAS action items involve providing information and software for the Parabolic Equation (PE) model and the ASeps TRansmission Loss (ASTRAL) model to SUBDEVRON and/or NUSC. This report documents the three action items involving the PE model. These action items are 1e, 1f and 1g. Sections 1-3 address these items. The action items involving ASTRAL are documented in Sections 4-6.

Action item 1e requests Parabolic Equation time arrival structure routines be delivered to NUSC for evaluation. The program needed for generation of time arrival structure is program BB, and is being delivered along with some possibly helpful ancillary software on an HP9020 floppy. The documentation of the time arrival structure routines is provided in this document in section 1.

Action item 1f addresses the delta-t ranging question. Section 2 of this report documents a method of delta-t ranging which should work well with both the PE and ASTRAL models in a range dependent environment.

Action item 1g requests an updated PE run time prediction for evaluation by SUBDEVRON. The updated prediction routine will be supplied on an HP9020 floppy, and the documentation of the changes made is contained in Section 3.

Action item 1c requests arbitrary beam pattern inputs for ASTRAL. This was easily provided in a non-configuration managed subroutine and the documentation is in section 4. Section 4 comprises the bulk of the PECP which will be presented at the next software review board (SRB) meeting. There was no HP9020 available to Mr. DeWayne White during the month in which he upgraded ASTRAL for this capability. Therefore, all routines needed to upgrade ASTRAL version 2.21 for the arbitrary beam pattern input are supplied on a PC floppy. Mr. White has been in contact with Steve Dolat of Sonalysts, who assured him that Sonalysts has the ability to transfer files between the PC and an HP9020.

Action item 1d for ASTRAL parallels item 1e for PE. The documentation of the ASTRAL time arrival structure is provided in section 5. There is a one-line change in the ASTRAL code needed, and this also is supplied on a PC floppy.

The documentation for Action item 1f is split between section 2, where a new delta-t ranging algorithm is suggested, and section 6, in which the ASTRAL interface with this algorithm is discussed.

Section 1: Item 1e: Provide Parabolic Equation (PE) travel time arrival structure routine to NUSC for evaluation. Generate ECP upon NUSC request.

The Parabolic Equation model is capable of producing arrival structure in time for broadband signals. This is made possible by the Fourier identity

$$P(t) = \mathcal{F}^{-1}\{\tilde{P}(f)\}$$

where

t is time,

f is frequency,

P(t) is complex pressure as a function of arrival time, and

\tilde{P} is complex pressure as a function of frequency.

In a single model run, the PE model produces and outputs complex pressure as a function of range and depth for a single frequency. Multiple runs of PE can produce complex pressure as a function of frequency for each range, depth pair of interest. These outputs from PE can then be used to produce, for a specific set of ranges and depths, complex pressure as a function of time. From P(t) then, intensity I(t) is generated by simply computing the square magnitude of the complex pressure. The name of the program used to generate travel times from PE output is program BB.

The method for performing a broadband PE run has been automated on the PC's, and could be automated similarly on an HP9020 or other computer. Figure 1 shows a flow chart of the processes involved in the broadband PE run. Each step in the process will be discussed.

Performing a Broadband PE run:

Step 1: Initializing Filename List

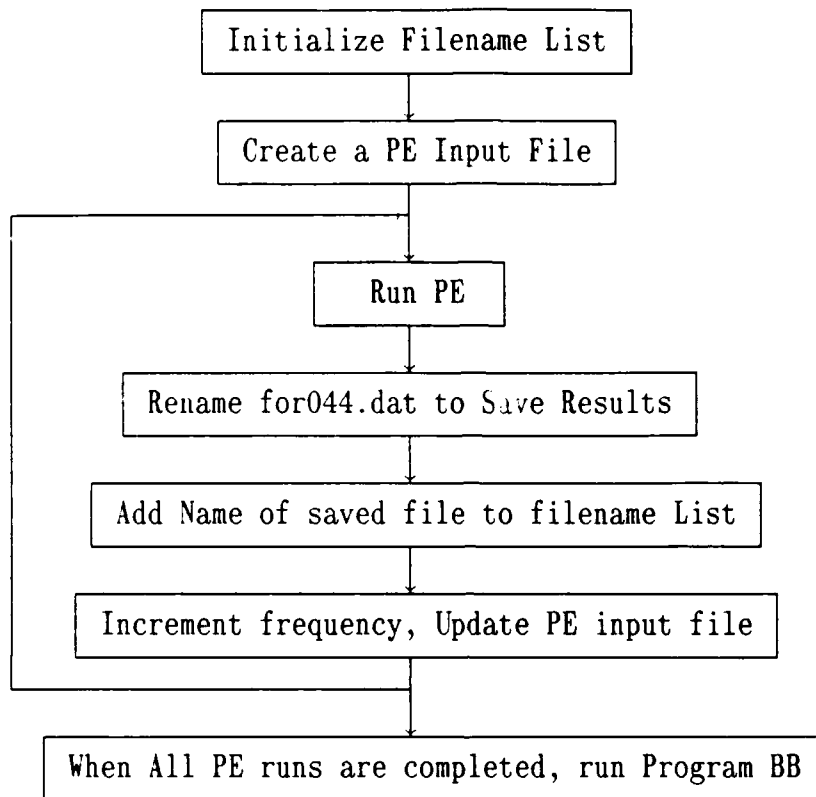
The broadband PE post processor, program BB, obtains some of its inputs from a file named **broadb.prc**. This file must be initialized at this time, and will be added to after each PE run. During this step, **broadb.prc** is opened as an ASCII file. A descriptive title for this broadband run is entered on line 1 of **broadb.prc**. The second line contains (format I5) the number of frequencies at which PE will be run. The format of this file is documented in Figure 2 at the end of this section.

Step 2: Creating a PE input file

Several steps are necessary to ensure that PE is saving the desired values from the complex pressure fields during execution. This section discusses specific inputs to PE important to the broadband processing algorithm.

Referring to the PE input set documentation, the flag NRBEAM (line 2, columns 56-60) must be set to indicate that the complex pressures will be saved and that other inputs are to follow on lines 2F and 2G. The depth window in which the complex pressures are to be saved is specified on Line 2F, and the ranges are specified on either Line 2F-2a or

Figure 1: Broadband PE Flow Chart



2F-2b, depending on whether NRBEAM is negative or positive. If NRBEAM is positive then NRBEAM ranges are input on Line 2F-2a. Otherwise the minimum range and range increment are given on Line 2F-2b.

It is essential that PE output complex pressure at *exactly* the same ranges, regardless of frequency. Therefore, to force PE to calculate the pressure field at the desired ranges, the target range option must also be requested through the model input file. To do this, the flag NUSDR (Line 2, columns 46-50) should be set to a negative value and the starting target range and target range increment (in nmi) should be input on line 2C-2.

Just as PE must output complex pressure at precisely the same set of ranges for each frequency, it also must output complex pressure at an identical set of depths. The flag NTRANS (Line 2, columns 26-30) should be set to the value of 1 to accomplish this. Setting the NTRANS flag will prevent the "shrinking" of the depth mesh which usually accompanies an increase in frequency. Instead, the depth mesh spacing is reduced when necessary by a factor of two, guaranteeing that every depth corresponding to a mesh point in the first (lowest frequency) PE run also corresponds to a mesh point in *all* PE runs.

Given a frequency band of interest, the frequency input on line 3 should be set to the minimum frequency of that band.

Step 3: Running PE

In this step, the user runs PE, using the appropriate PE input file. For the first PE run of a broadband run, the input file contains the frequency corresponding to the lower edge of the frequency band of interest. For each subsequent run, the frequency is incremented until the entire frequency band has been covered. Each PE run produces one output file of interest, and that file is **for044.dat**, which contains the complex pressures in the desired depth window at the desired ranges. Step four saves this file so that it is not over-written during the next PE run.

Step 4: Renaming the Complex Pressure File

On the PC, after each PE run the file **for044.dat** is renamed **pennnnn.044** where nnnnn is the truncated integer [frequency*100]. This is just an illustration of one method of file naming which obviously is not usable for frequencies above 1000 Hz or frequency increments less than .01 Hz. It serves only as an example of a scheme by which each file output from PE is given a unique name.

Step 5: Adding Name of Saved File to Filename List

The post processor, program BB, reads from the filename list, **broadb.prc**, to obtain the names of the complex pressure files output from PE. Therefore, this step is the logical place in the flow chart to save the name of the last complex pressure file created.

Step 6: Updating the PE File for the Next Frequency

A simple program has been written for the PC which reads in a PE input file, increments the frequency and copies the contents to a new file. This program can be converted to any other computer easily, or the implementer may wish to write his own

version.

A remaining question may be the choice of frequency increment. For the case when outputs are to be used for delta-t ranging, the frequency increment is discussed thoroughly in the documentation for item 1f. However, if the user would like simply to observe predicted arrival structure, there is a simple algorithm for choosing the frequency increment. First, if the total length T (in time) of the arrival is known, the frequency increment should be set to $1/(2T)$. The factor of 2 in the denominator will help to avoid any ambiguity caused by the FFT wraparound properties (see Step 7). If the total length of the arrival is not known, T can be estimated in a very conservative fashion by setting

$$T = T_{\max} - T_{\min}$$

with

$$T_{\min} = R/C_{\max}$$

and

$$T_{\max} = (R/\cos(\theta))/C_{\min}$$

where

R is the total range from source to receiver,

C_{\max} is the maximum sound speed in range and depth along the track,

C_{\min} is the minimum sound speed in range and depth along the track, and

θ is the PE vertical beamwidth.

In other words, sound cannot propagate between source and receiver any faster than at the maximum sound speed, in a straight line. Also, sound cannot propagate slower than at the steepest possible angle and the slowest sound speed.

Step 7: Running the Post Processor

When all PE runs have been performed, the frequency loop is exited, and program BB is run. This program reads inputs from three sources. The first source is the file **broadb.prc**, which contains the Title for this broadband PE run, the number of frequencies, and the filename of the PE complex pressure file for each frequency.

Next, BB reads the first (lowest frequency) PE complex pressure file and presents the user with choices of depths and ranges of interest. The user enters the depths and ranges of interest interactively from the console, after which program BB loops through all PE complex pressure extracting the complex pressure at the user-specified set of ranges and depths. Complex pressure as a function of frequency is then inverted using the FFT to complex pressure as a function of time. This inversion is performed for each requested range and depth combination. Finally, intensity in dB as a function of time is computed and output to the file **bb.out**. The format of this file is described in Figure 3.

A note should be made at this point about the FFT wraparound property. Since the broadband PE algorithm provides relative travel time and not absolute travel time, there is always an ambiguity in travel time. For a single range r_i and depth z_j , the output from program BB is

$$I(r_i, z_j, t)$$

where I is intensity and t is relative time in seconds. However, to carefully define relative time, we must say that absolute time t_{abs} is defined

$$t_{abs} = t_0 + t + nT \quad \text{for } n=0,1,2,\dots$$

where t_0 is an arbitrary constant and T is $1/\Delta f$, Δf being the frequency increment. This is a direct result of the assumption by the Fourier transform that all functions are harmonic. In this equation, n is an unknown, but because we have chosen T large enough to encompass all arrivals, the relative arrival structure should be clear. As an example, consider a case where t_0 is 0, T is 4 seconds, and our scenario is such that arrivals are expected at 7.4 and 9 seconds. In this case, the function $I(r_i, z_j, t)$ will have peaks at relative times $t=3.4$ and $t=1$. For simplicity this property is not mentioned again in the discussion of delta- t ranging, but we will state here that both methods of performing the correlation are not negatively affected by this property. In fact, the FFT correlation depends on the wraparound principle, and the direct method is easy to implement with the required wraparound.

Figure 2: File format for broadb.prc, an Input file for Program BB

File Name: **broadb.prc** (must be lower case on UNIX systems)

File Characteristics: Formatted, ASCII file
2 Header Records
Variable number of Data records

<u>Record #</u>	<u>Format</u>	<u>Contents</u>	<u>Description</u>
1	A32	Title	ASCII Descriptive information
2	I5	NFREQ	Number of frequencies at which PE is run
3-NFREQ+2	A32	FILE	Name of file containing complex pressures PE run. Record 3 contains the name of the file containing complex pressures from the first PE run, record 4 contains file name from second PE run, etc.

File Name: bb.out (must be lower case on UNIX systems)
File Characteristics/Organization: Formatted, ASCII file

3 Header Records

Range Records

<u>Record type</u>	<u>Format</u>	<u>Item</u>	<u>Description</u>
1	A32	Title	ASCII Descriptive information
2	3I5,F10.6	NR,NT,NZ,DT	<p>NR = Number of ranges at which intensity vs time is provided</p> <p>NT = Number of time points in the output intensity vs time array</p> <p>NZ = Number of depths at which intensity vs time is provided</p> <p>DT is the increment in travel time for the intensity vs time output arrays</p>
3	8F10.2	(Z(J),J=1,NZ)	Array of depths in feet at which intensity vs time is provided
4	F10.2	R(I)	I'th range in Nmi. This record acts as a sub-header record for a block of data records for this particular range. Record 4 and records 4a are repeated for each of the NR ranges for which intensity vs time is output
4a	8F10.2	(TL(R(I),Z(J),TIME(K),K=1,NT)	Array of intensity (expressed as dB) vs time for the J'th output depth, Z(J). This record is repeated within record 4 for each output depth.

Section 2: Item 1f: Provide NUSC with PE and ASTRAL multipath ranging information to support delta-T ranging algorithm testing and implementation.

The algorithm SAIC suggests for delta-t ranging using Parabolic Equation outputs should closely resemble the algorithm currently used in the APP program, with the exception that the identification of individual arrivals is not needed. The ability to identify paths as "bottom bounce", RSR, etc, is questionable in a range dependent environment, and while individual rays can be identified with reflections off boundaries, this identification can be extremely complicated. Not only the total number of bottom bounces, bottom horizontals, surface reflections and surface horizontals, but also the sequence of each of these "events" must be remembered. Speaking of a path as CZ can be misleading if that path converts to bottom bounce on an upslope. The method presented here for delta-T ranging should be robust when used in a complicated, range dependent environment.

As discussed in Item 2e, the intensity output from the broadband PE algorithm is defined on a range, depth and time grid:

$$I_{pe}(r_i, z_j, t)$$

where

I_{pe} is the intensity predicted by the Parabolic equation model.

r is range,

z is depth, and

t is relative travel time.

We wish to find the specific range and depth combination (r_c, z_c) where the function $I_{pe}(r_c, z_c, t)$ most closely correlates with some incoming data $I_d(t)$. An efficient way of computing r_c and z_c is to form a two-dimensional matrix

$$C_p(r, z)$$

where

$C(r_i, z_j)$ is the level of the peak of the correlation function

$$I_d(t) * I_{pe}(r_i, z_j, t)$$

where the * symbol denotes the correlation. Thus,

$$C_p(r_i, z_j) = \text{MAX}_{\tau} (C(r_i, z_j, \tau))$$

$$C(r_i, z_j, \tau) = \sum_t I_{pe}(r_i, z_j, t) I_d(t - \tau)$$

where the correlation delay parameter τ is defined on the same mesh as is time t . We will call this method of correlation the direct method. Of course, a faster way of performing the correlation utilizes the Fourier relationship

$$\mathcal{F}\{A * B\} = \mathcal{F}\{A\} \mathcal{F}\{B\}$$

which states that the Fourier transform of the correlation of two functions is the product of

the Fourier transform of the first function and the Complex conjugate of the Fourier transform of the second function. This is actually just an extension of the identity relating convolution with multiplication. To utilize the Fourier relationship, one performs a Fourier transform and conjugation on incoming data $I_d(t)$ only once, to define

$$\tilde{I}_d(f) = \mathcal{F}\{I_d(t)\}^*$$

where f is frequency, then for each range, depth pair of interest, this result is multiplied by

$$\tilde{I}_{pe}(r_i, z_j, f) = \mathcal{F}\{I_{pe}(r_i, z_j, t)\}$$

and the product is inverted producing

$$C(r_i, z_j, \tau) = \mathcal{F}^{-1}\{\tilde{I}_{pe}(r_i, z_j, f) \tilde{I}_d(f)\}$$

A simple analysis of computation times of the two methods shows that the Fourier method of correlation should improve performance vastly. Consider a matrix output from the broadband PE post processor whose dimensions are l -ranges, m -depths and n -times. The number of multiplications N_1 needed in the first method to compute $C(r, z, \tau)$ from $I_{pe}(r, z, t)$ and $I_d(t)$ is

$$N_1 = l * m * n * n$$

where right side of the equation is the number of multiplications needed for the direct correlation method. The number of multiplications N_2 needed for the Fourier method is

$$N_2 = l * m * n + (2 * l * m + 1) * n * \log_2(n)$$

The first term in this equation represents the multiplication in the frequency domain and the second term represents the number of multiplications needed for the forward and inverse Fourier transforms, including the single forward transform of the data. The following table illustrates the advantage of the Fourier method for some concrete examples. Note that N_1 and N_2 are both linearly dependent on the value $l * n$, so only one typical value was picked for the table.

Table 1: Relative Number of multiplications for the direct and Fourier methods of correlation processing.

$l * m$	n	N_1	N_2
300	32	3.1×10^5	1.1×10^5
300	64	1.2×10^6	2.5×10^5
300	128	4.9×10^6	5.8×10^5
300	256	2.0×10^7	1.3×10^6

An issue not addressed so far is the choice of frequency band and frequency interval used in the PE runs. With a given a sample of input data $I_i(t)$, a sampling rate Δt_i and duration T is defined. The frequency band of the incoming signal is also known. Define f_1 and f_2 as the lower and upper limits of the frequency band of the incoming signal, respectively. To efficiently use the correlation method described above, we need the output time series from PE to be defined on the same time grid as is the input data. Define

$$F_{\max} = 1/\Delta t_i$$

and

$$F_d = f_2 - f_1$$

Here, F_{\max} is the largest band for which the sample rate Δt_i is sufficiently dense so as not to alias. If F_d is larger than F_{\max} , the sample rate for the incoming data must be increased. We will now assume $F_d \leq F_{\max}$. The ratio F_{\max}/F_d gives us an idea of the amount of over-sampling in time, and the time series from the incoming signal should be decimated as much as possible. The decimation factor N_d is defined as

$$N_d = \left\lfloor \frac{F_{\max}}{F_d} \right\rfloor$$

where $\lfloor x \rfloor$ denotes the integral part of x (and since x is positive, the largest integer less than or equal to x). Now the array $I_d(t)$ is simply the incoming sampled data $I_i(t)$ decimated by a factor of N_d . The new sampling frequency, Δt_d is defined

$$\Delta t_d = N_d \Delta t_i$$

The decimated time series, $I_d(t)$ has an effective maximum bandwidth F_{eff} of $1/\Delta t_d$. With $I_d(t)$ defined we are ready to define the frequencies at which the PE predictions are made.

We have two criteria which must be met in our definition of Δf , the frequency spacing of the PE predictions. First,

$$\Delta f \leq 1/T$$

ensures the time series output from the broadband PE algorithm will have a sufficiently long duration. Secondly,

$$\Delta f = \frac{F_{\text{eff}}}{2^N}$$

for some integer N . This allows the broadband algorithm to use an FFT to perform the Fourier transform used to compute $I_{pe}(r_i, z_j, t)$. Combining these restrictions, we have

$$F_{\text{eff}} T \leq 2^N$$

and choose

$$N = \lceil \log_2(F_{\text{eff}} T) \rceil$$

where $\lceil x \rceil$ is used here to denote the smallest integer not less than x . Now that we have

defined the frequency spacing, we need only define the starting frequency and number of frequencies for the PE predictions which will provide complex pressure as a function of range, depth and frequency $P(r_i, z_j, f)$ to the broadband PE algorithm. Since the broadband algorithm takes complex pressure $P(r_i, z_j, f)$ and zero-fills in frequency to a power of 2, there is not necessarily any need to perform 2^N PE runs. The number N_{pe} of PE runs needed depends on the signal bandwidth, F_d .

$$N_{pe} = \lceil \frac{F_d}{\Delta f} \rceil$$

is the number of PE runs needed. The starting frequency for the PE predictions is f_1 , the frequency at the lower edge of the signal frequency band.

The following table contains examples of this computation sequence.

Table 2: PE run parameters and sample data decimation rates given sample input data parameters.

Input Data Parameters				Intermediate Results			PE run Parameters	
Δt_i	T	f_1	f_2	N_d	Δt_d	2^N	Δf	N_{pe}
.001	2	380	420	25	.025	128	.3125	128
.001	2	360	395	28	.028	128	.279	126
.001	1.5	360	395	28	.028	64	.558	63
.001	1.5	300	450	6	.006	256	.651	231
.001	10	299	301	500	.5	32	.0625	32

Item 2g: Provide updated "SWITCH" PE run time predictor to COMSUBDEVRON TWELVE for evaluation.

One of the criteria for making a choice of transmission loss model is that of run time. In many situations, the PE model is too slow to be run. Therefore, it is desirable to know the amount of time needed for a model run in a given environment. SAIC has provided run time estimates for the PE and ASTRAL models. The ASTRAL model run time is fairly accurate and is a simple function of the number of source and receiver depths and the number of range points. This algorithm has been tested and found satisfactory. PE run time estimation is much more difficult, given that the PE model uses a variable range step to produce very accurate results as efficiently as possible. The initial version of the PE run time estimate consisted of a simple, range independent ray trace. Along a single ray launched from the starting point at the maximum vertical beam width, the expected PE range step was computed, and averaged for the entire length of the PE track. The total number of range steps expected for the PE run was estimated in this way, and multiplied by the average CPU time used per range step to produce the predicted run time. The SAIC technical note, ESTIMATING PE RUN TIME, is enclosed.

COMSUBDEVRON TWELVE was provided with this first attempt at a PE run time prediction, and found that in bottom limited cases, the run time estimation grossly under-predicted the actual run time. Discrepancies as large as factors of 10 were found. We have therefore upgraded the part of the run time estimation which predicts the number of range steps in a given PE run. In the bottom limited case, it is assumed that there will be energy at all angles within the PE vertical beam, at all ranges. In this case, the ray trace mentioned above is not run, but the predicted range step is based on the maximum angle of propagation, the total PE vertical beam width. Results for a small number of test cases show that this algorithm over-predicts the PE run time in some cases, but never under predicts run times. The over-prediction in run time averages 20% and was never more than 50%, a great improvement over the 1000% error in the earlier versions.

The PE run time prediction branches into its bottom limited mode whenever the sound speed at the bottom is less than or equal to the sound speed at the fixed PE input depth. In this case, the estimated PE range step is defined as

$$\Delta r_{\text{est}} = \frac{\lambda}{1 - \cos(\theta)}$$

where λ is the acoustic wavelength and θ is the maximum angle of propagation. The actual range step used in the PE model is

$$\Delta r_{\text{pe}} = \frac{\lambda}{1 - \cos(\theta_{\text{curr}})}$$

where θ_{curr} is the maximum angle at which any significant energy is propagating at the current range. The PE run time estimator cannot quickly compute θ_{curr} , so this must be estimated. Three different estimates of θ for the bottom limited environment were tested.

First, the effective beamwidth θ_{eff} of PE is computed,

$$\theta_{\text{eff}} = \cos^{-1}\left(-\frac{c_{\text{bot}}}{c_{\text{src}}} \cos(\theta_{\text{pe}})\right)$$

where c_{bot} and c_{src} are the sound speeds at the bottom and PE input depth respectively and θ_{pe} is the input PE vertical beamwidth. θ_{eff} is computed using Snell's law from the PE input depth to the bottom.

The second candidate angle is computed from θ_{eff} by considering the PE filter rolloff function. In the PE model, the filter rolloff in angle space attenuates energy traveling at angles steeper than θ_{eff} so that no energy remains at an angle of θ_{filt} where

$$\theta_{\text{filt}} = \sin^{-1}(4/3 \sin \theta_{\text{eff}})$$

The filter function is constant with a level of 1 between angles of 0 and θ_{eff} , and rolls off smoothly to a level of 0 at an angle of θ_{filt} . Clearly, there should never be energy propagating at angles greater than θ_{filt} . This is expected to produce a very conservative run time estimate.

The third candidate angle is a compromise between θ_{eff} and θ_{filt} . The angle $\theta_{3\text{db}}$ at which the filter function has a level of 0.5 (or "3 dB down") is

$$\theta_{3\text{db}} = \sin^{-1}(1.2 \sin(\theta_{\text{eff}}))$$

and this turns out from experimentation to be a fairly good estimate of θ_{curr} .

Four test cases, all bottom limited environments, were run to test the accuracy of the modification to the PE run time estimation. For all cases, the water column was 3000 feet deep, the maximum range was 7 Nmi, the frequency was 100 Hz and the sound speed profile consisted of a single layer with a constant negative gradient. Table 3 shows some PE run time statistics for each of the four test cases.

Table 3: PE run time statistics as a function of gradient

	Case Number			
	1	2	3	4
Sound speed gradient	-.003	-.067	-.100	-.333
PE effective beamwidth(deg)	20.1	22.7	24.0	32.5
Average Range Step (ft)	735	556	486	223
Number of range steps	58	77	87	190
PE run time in seconds	8.7	17.5	19.9	43.5

Note: The PE effective beamwidth is θ_{eff} defined above

Table 4 compares the performance of the PE run time estimation in these four bottom limited cases for the three possible values of θ used in predicting the PE range step. Using the first choice for θ , the run time estimation under-predicts the number of range steps for all cases except the first. As we expected, using θ_{filt} in the range step prediction is too conservative. In this case, all predictions of the number of range steps taken were far too large. The choice of θ_{3db} seems to be the best yet. This choice shows a smaller margin of error than either of the previous two while never under-estimating the number of steps taken in the PE run.

Table 4: PE number of range step predictions for three candidate formulas compared to actual number of range steps taken

<u>Case #</u>	<u>Actual</u>	Formula using angle of		
		<u>θ_{eff}</u>	<u>θ_{filt}</u>	<u>θ_{3db}</u>
1	55	58	106	85
2	93	77	141	112
3	117	87	162	129
4	247	190	367	286

Section 4: Action Item 1c: Verify ASTRAL beam pattern capability; generate any ECP's necessary to provide this capability under configuration controlled ASTRAL.

Subroutines RCVINP and BEAMGN were modified to read in and define the initial mode amplitudes for user-specified beampattern. See Appendix A for programming details. The non configuration managed subroutine RCVINP was changed to accept three different inputs for the IRSYS flag:

IRSYS =1 (as previously this indicates a VLA system as explained in "OAML-SPS-23: Software Product Specification for the ASTRAL 2.2 Model", Vol. 1, Apr. 1988, p 501). Also see pp 525 and 526. Note that the "(IRCV=0)" at the top of page 525 (in the NOTE:) should be a "(IRCV=1)".

=2 This is a new option. When IRSYS is input as 2, a user beam pattern set is expected in the following form:

Line Type 16A: INFPRF, NFB

Format 2I5

Index of the first SSP on the track (i.e. SSP index in which the receiver is sitting) and number of frequencies for which a beam pattern input is to made.

*** LINES 17A and 18A are repeated for each frequency ***

Line Type 17A: FREQBP(M), IBPTYP(M)

Format F10.2,I5

Frequency for the beam pattern and the number of receiver angles at which the beam pattern is to be input. The sign of IBPTYP depends on the 'type of beam pattern input', i.e. if the beampattern level is to be input as dB, IBPTYP(M) should be negative and if an intensity factor is to be input, IBPTYP(M) should be positive.

Line Type 18A:

{(THUPDN(M,I),BPUPDN(M,I),I=1,NN),M=1,NFB}

Format 8F10.2

The NN above is the absolute value of IBPTYP(M). THUPDN(M,I) is the Ith receiver angle in degrees, negative going up toward the surface. BPUPDN(M,I) is the receiver response or the beampattern level at that angle (either dB or intensity according to the sign of IBPTYP(M)). Note THUPDN is assumed to be monotonically increasing and BPUPDN is constant for angles less than THUPDN(M,1) [at BPUPDN(M,1)] and greater than THUPDN(M,NN) [at BPUPDN(M,NN)].

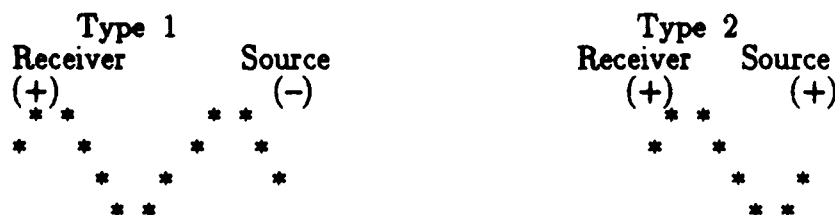
Note that although the beam pattern input will be read if IRSYS is equal to 2, it will only be used if, in addition, there is a suspended receiver (IRVC = 1), surface image interference is requested (IRDE = 1) and there is a flat near field bottom (ISLOPE = 1).

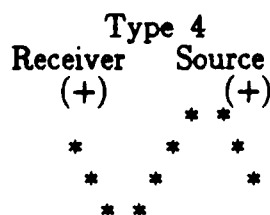
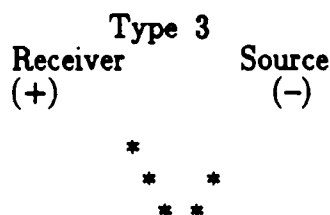
The SPS ["OAML-SPS-23: Software Product Specification for the ASTRAL 2.2 Model", Vol. 1, Apr. 1988 by D. White, L. Dozier, and C. Pearson] briefly discusses the ASTRAL arrival time structure output on page 510 and page 530 in the discussion of DEBUG(3). On page 530 in the discussion of the input variable DEBUG(3), it is indicated that the structure of the arrival time output files is written to unit NOUT. Following is a more complete explanation of these files.

$$xx = 56 + N + 3 * M.$$

Source	Frequency	File unit	File name
200	50	60	FOR060.DAT
300	50	61	FOR061.DAT
400	50	62	FOR062.DAT
200	100	63	FOR063.DAT
300	100	64	FOR064.DAT
400	100	65	FOR065.DAT
200	150	66	FOR066.DAT
300	150	67	FOR067.DAT
400	150	68	FOR068.DAT

ZONE 1 RAYS FOR A GIVEN ANGLE





Thus at each zone and for each range we have the set of quantities

RANGE, TRAVEL TIME, I1, I2, I3, I4, THS, THR

where I1 corresponds to a —+ ray (Type 1) intensity, I2 to a ++ ray (Type 2), I3 to a +- ray (Type 3), and I4 corresponds to a — ray (Type 4), the THS/THR is the magnitude of the angle at the source/receiver and the corresponding sign of the angle is given by the arrival type, i.e. the —+ ray leaves the source going up (negative angle) and arrives at the receiver going down (positive angle). An angle of zero indicates a 'diffractive arrival'. These quantities, in this order, correspond to a record of the arrival time structure for ASTRAL. The records will not necessarily be in order of increasing range. The intensities are relative so that to obtain transmission loss (TL) in dB, the formula

$$TL = -10\log(\text{Intensity}) + 28.29$$

is used, where 28.29 is a normalization factor.

Section 6: Action Item 1f-ASTRAL: Provide NUSC with ASTRAL multipath ranging information to support delta t ranging algorithm testing and implementation.

In discussions with Steve Dolat, DeWayne White was able to conclude that the particular information provided by ASTRAL and/or PE need not have the same structural arrival time information as that which RAYMODE provides. According to Mr. White's conversations with Steve Dolat, the only information available from actual data is an azimuth, sound pressure level, and ('wall clock') time of arrival. Given this available information, another method of using the predicted arrival time information to obtain ranging information has been suggested by E. Holmes of SAIC (see section 2).

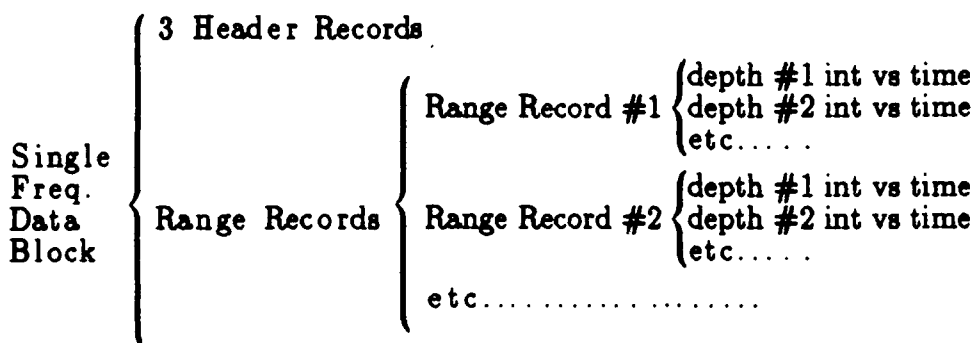
Although the information supplied in Section 5 should be sufficient for the 'usual delta t ranging algorithm', Mr. White has supplied a (non-configuration managed) program which will build an (ordered) range, travel time, level(dB) file from the ASTRAL arrival structure files.

This program, ASTRALTT, has as its input the output file from ASTRAL, "OUTPUT.DAT", [see p 529 of the SPS volume 1], the bandwidth, BW, and the time period, T, of the processing system as well as the FOR0xx.DAT ASTRAL arrival structure files, described in Section 5. The "OUTPUT.DAT" file is used only to obtain the number of frequencies, NF, number of source depths, NZ, and number of ranges, NR, (and thus could be replaced with a file containing only this information). The input bandwidth, BW, is used to 'bin' the travel time output into NT (equal to BW times T) time bins which are DELAT (equal to one over BW) wide. Times are scaled to zero for each range bin (i.e. the first arrival time for each range is subtracted from the other times).

The output of ASTRALTT is the formatted file ASTTT.DAT which has the same basic format as the PE arrival structure file. However, since the information is available at multiple frequency bands as well as at multiple depths, ASTTT.DAT contains output information for all frequency bands. The output format for this file is described in Figure 4.

Figure 4: File format for asttt.dat, The ASCII Output file from Program ASTTT

File Name: asttt.dat (must be lower case on UNIX systems)
 File Characteristics/Organization: Formatted, ASCII file



<u>Record type</u>	<u>Format</u>	<u>Item</u>	<u>Description</u>
1	A32	Title	Title is ASCII Descriptive information
2	3I5,2F10.3	NR,NT,NZ,DT,F	NR = Number of ranges at which intensity vs time is provided NT = Number of time points in the output intensity vs time array NZ = Number of depths at which intensity vs time is provided DT is the increment in travel time for the intensity vs time output arrays F is frequency, unless it is set ≤ 0.0 , in which case F is a flag indicating an End of file, this being the last record in the file.
3	8F10.2	(Z(J),J=1,NZ)	Array of depths in feet at which intensity vs time is provided
4	F10.2	R(I)	I'th range in Nmi. This record acts as a sub-header record for a block of data records for this particular range. Record 4 and records 4a are repeated for each of the NR ranges for which intensity vs time is output
4a	8F10.2	(TL(R(I),Z(J),TIME(K),K=1,NT)	Array of intensity (expressed as dB) vs time for the J'th output depth, Z(J). This record is repeated within record 4 for each output depth.

For each frequency of interest, record types 1 through 4 are repeated. After the last frequency of interest, this file repeats records 1 and 2, with the frequency in record type 2 being set to 0.0.

Appendix A

ASTRAL Programming Notes

As mentioned in Section 4, subroutines BEAMGN and RCVINP have been modified for the beam pattern input. Subroutine RCVINP, currently non-configuration managed, reads the beam pattern input into COMMON blocks. Subroutine BEAMGN, under configuration mangement, computes the initial mode amplitudes given the beam pattern level as a function of angle.

COMMON block ADLINP contains the new arrays associated with the user input beam pattern, therefore other routines containing this COMMON block will need to be recompiled before creating the executable. Following is a list of subroutines which contain COMMON block ADLINP.

- astdrv.f
- beamgn.f
- blockd.f
- trkinp.f
- rcvinp.f

Subroutines BEAMGN and RCVINP and "INCLUDE" file ADLINP.COM are being provided on a PC-formatted floppy.